Session 139

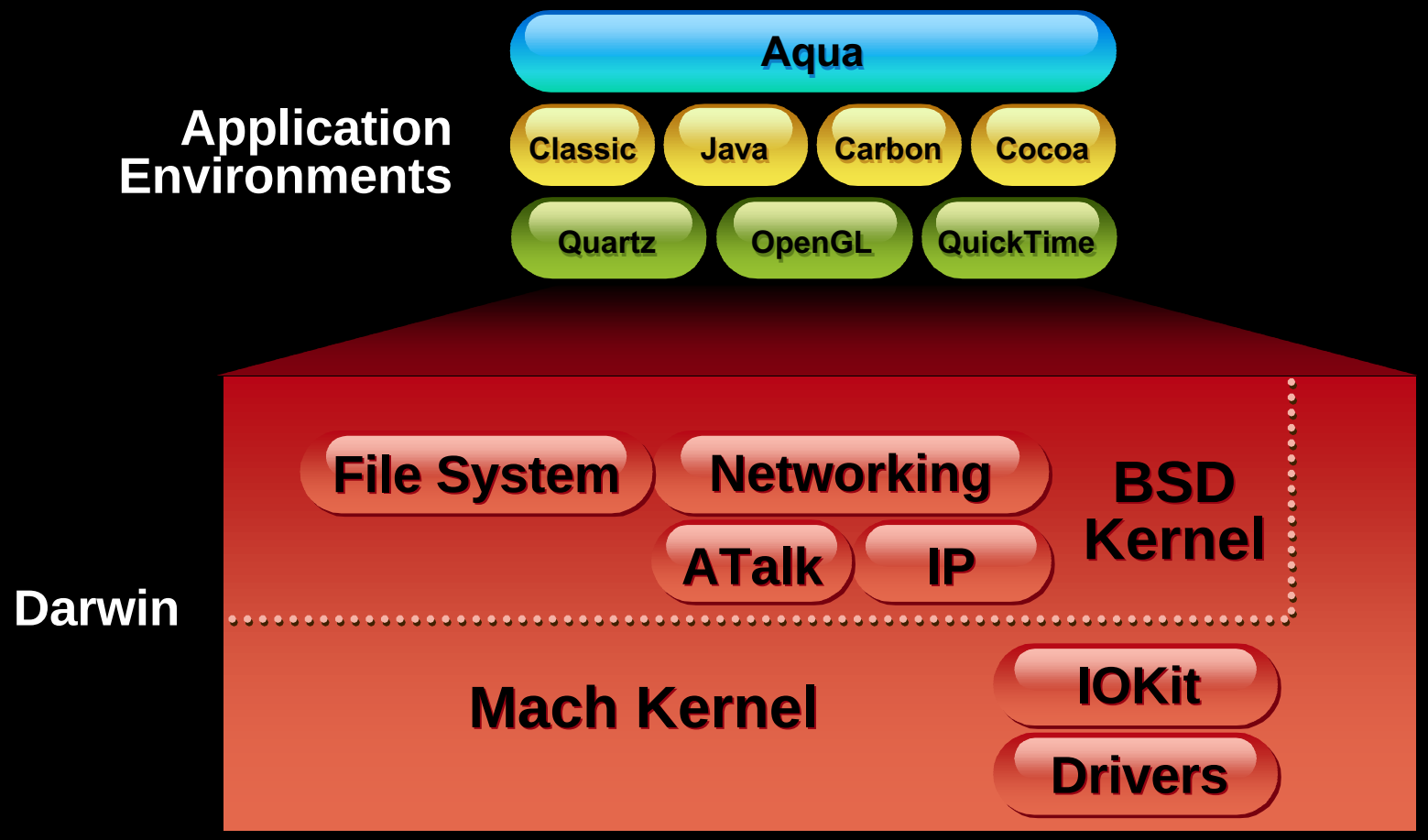
Leveraging BSD Services in Mac OS X

Brett R. Halle
Director, Core OS Engineering

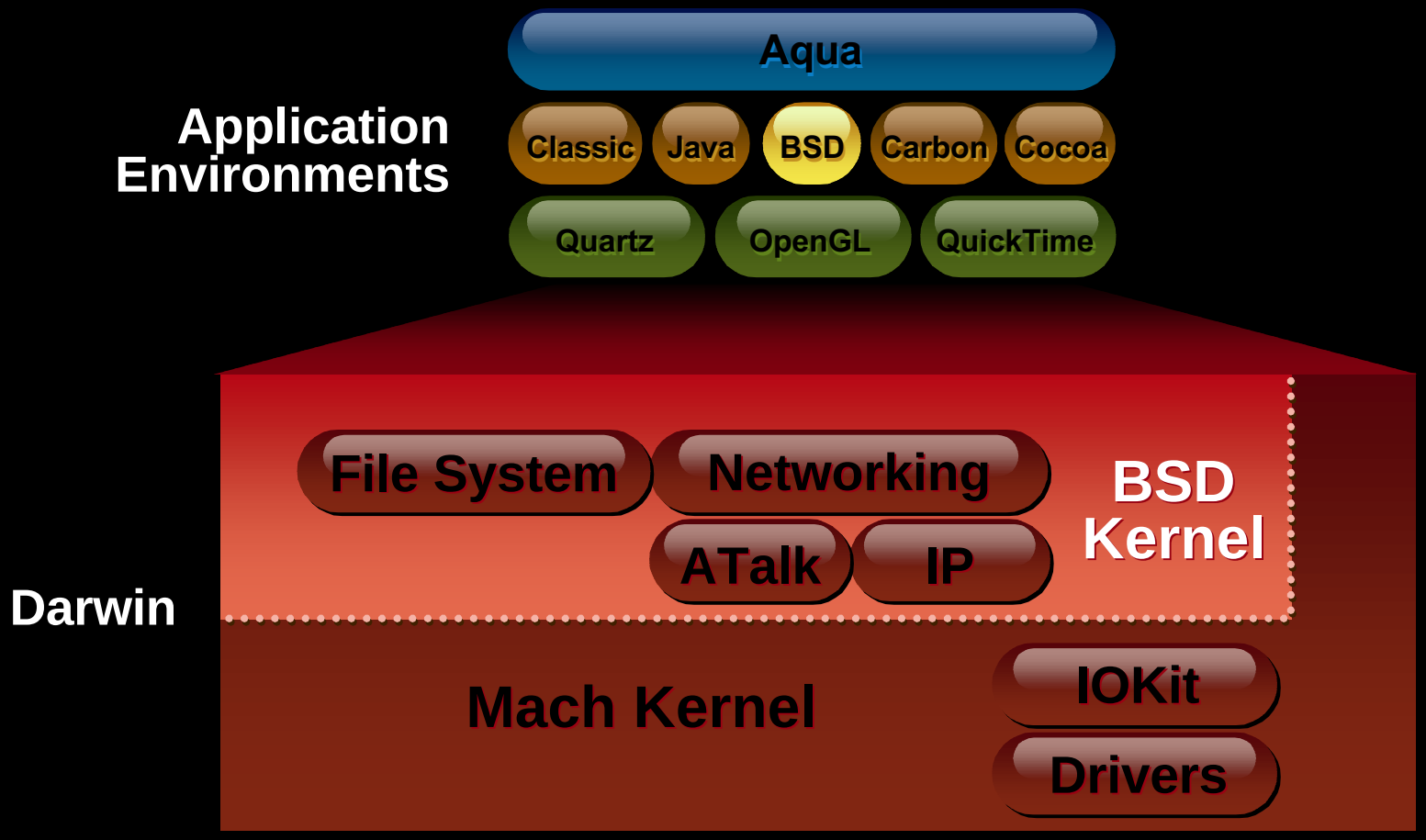
Topics

- What is BSD?
- BSD's role in Mac OS X
- Programming with BSD
- The BSD community

Darwin/Core OS



BSD: The Power of UNIX



The Darwin Kernel

- BSD sits on top
  - File system layer
  - Network stack
  - POSIX functionality
- I/O Kit owns the driver model
- Mach sits underneath
  - Tasks and threads
  - Memory
  - Manages the processor

What Is BSD?

- It's an Operating System!
  - The BSD kernel
  - The BSD libraries
  - The BSD application environment

BSD Kernel

- Based on BSD 4.4 plus integration w/Mach and I/O Kit
- Provides “OS Personality” APIs and Services
  - Process model
  - Basic security policy
  - File system architecture
  - Networking architecture

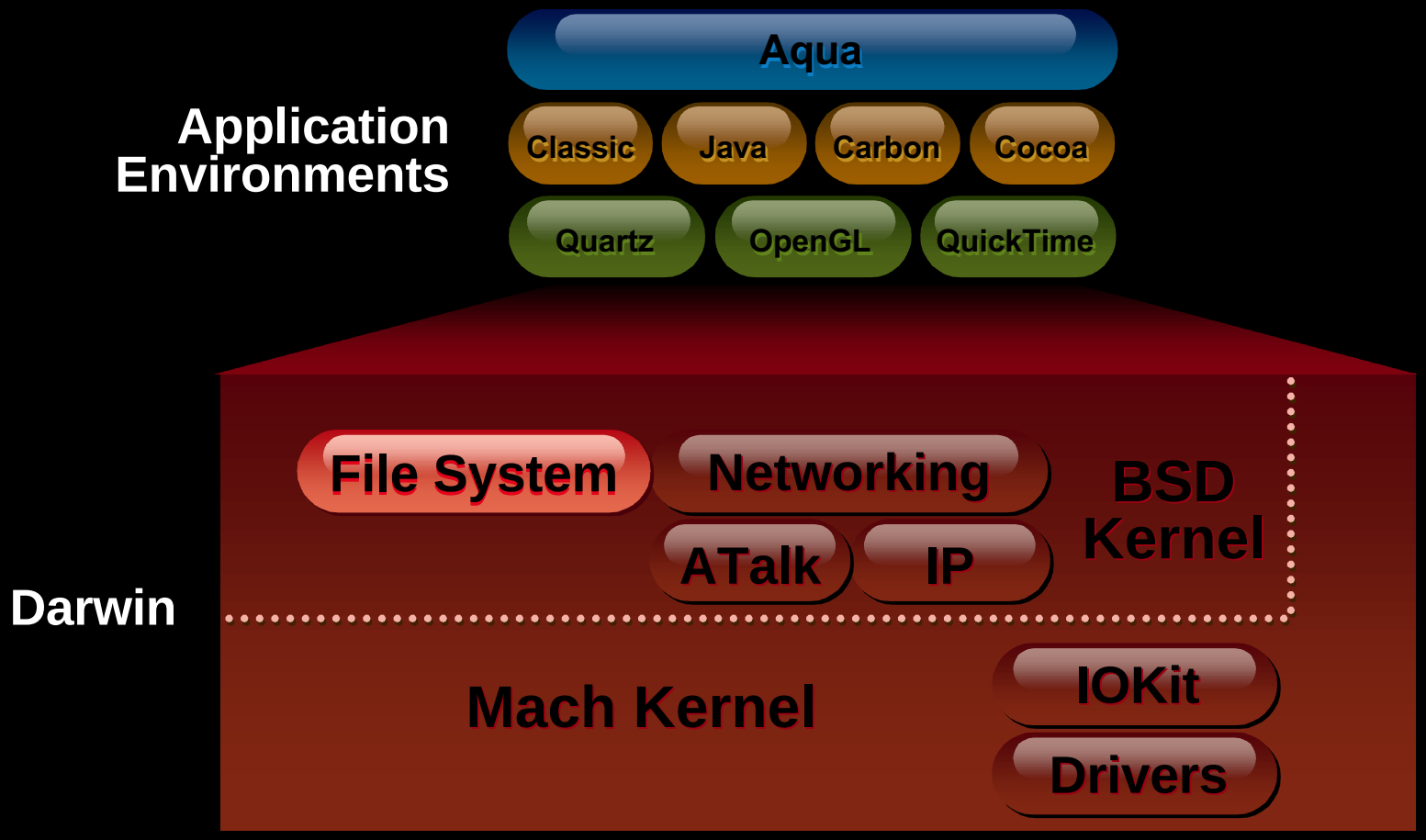
Process Model

- BSD Process vs. Mach Task/Thread
- OS resource management
  - File descriptors
  - Memory
  - Network resources
- Environment variables
- Signal delivery

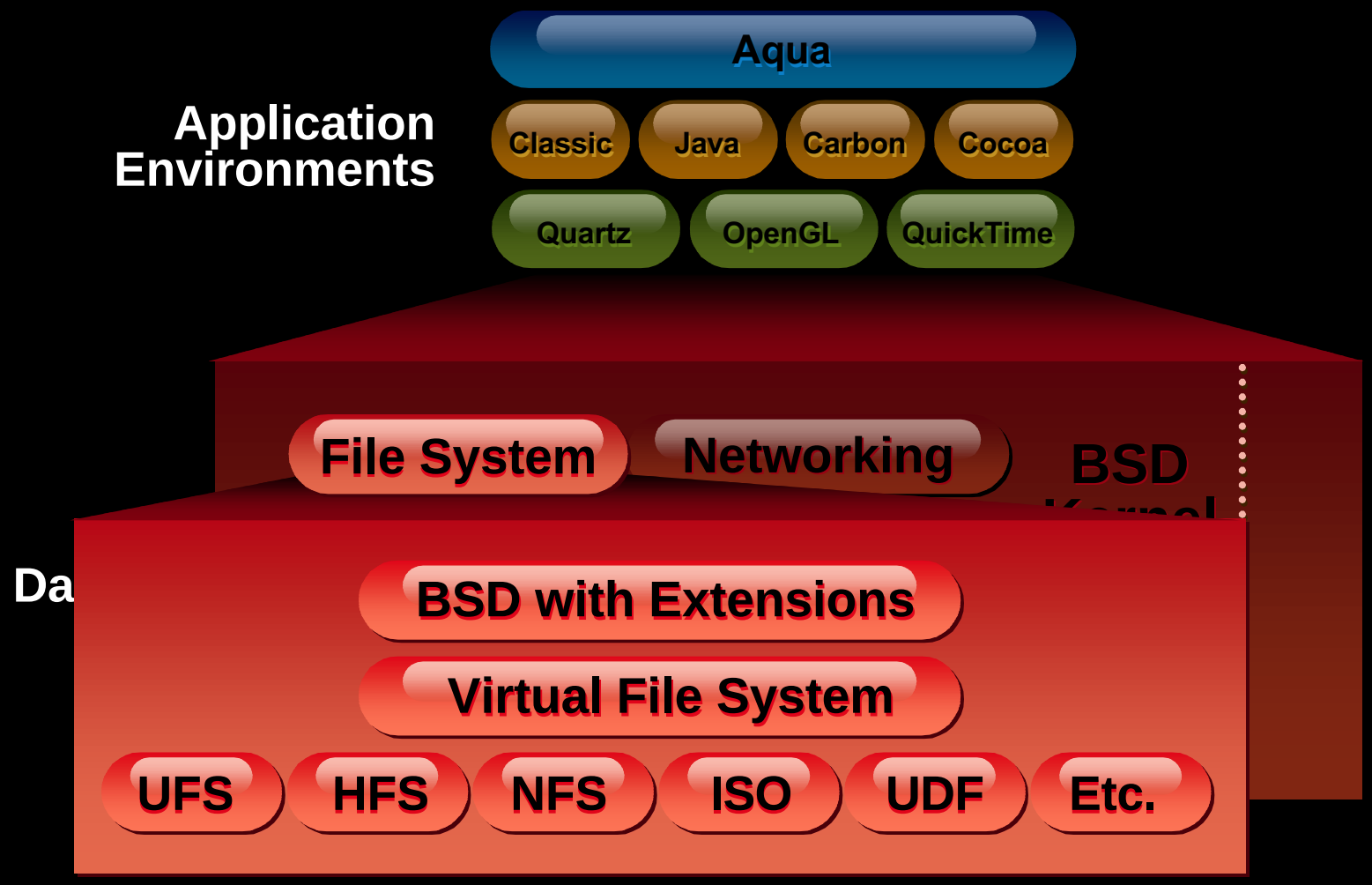
Security Policy

- “Users”
- Super user vs. everyone else
- File system access
  - Users
  - Groups

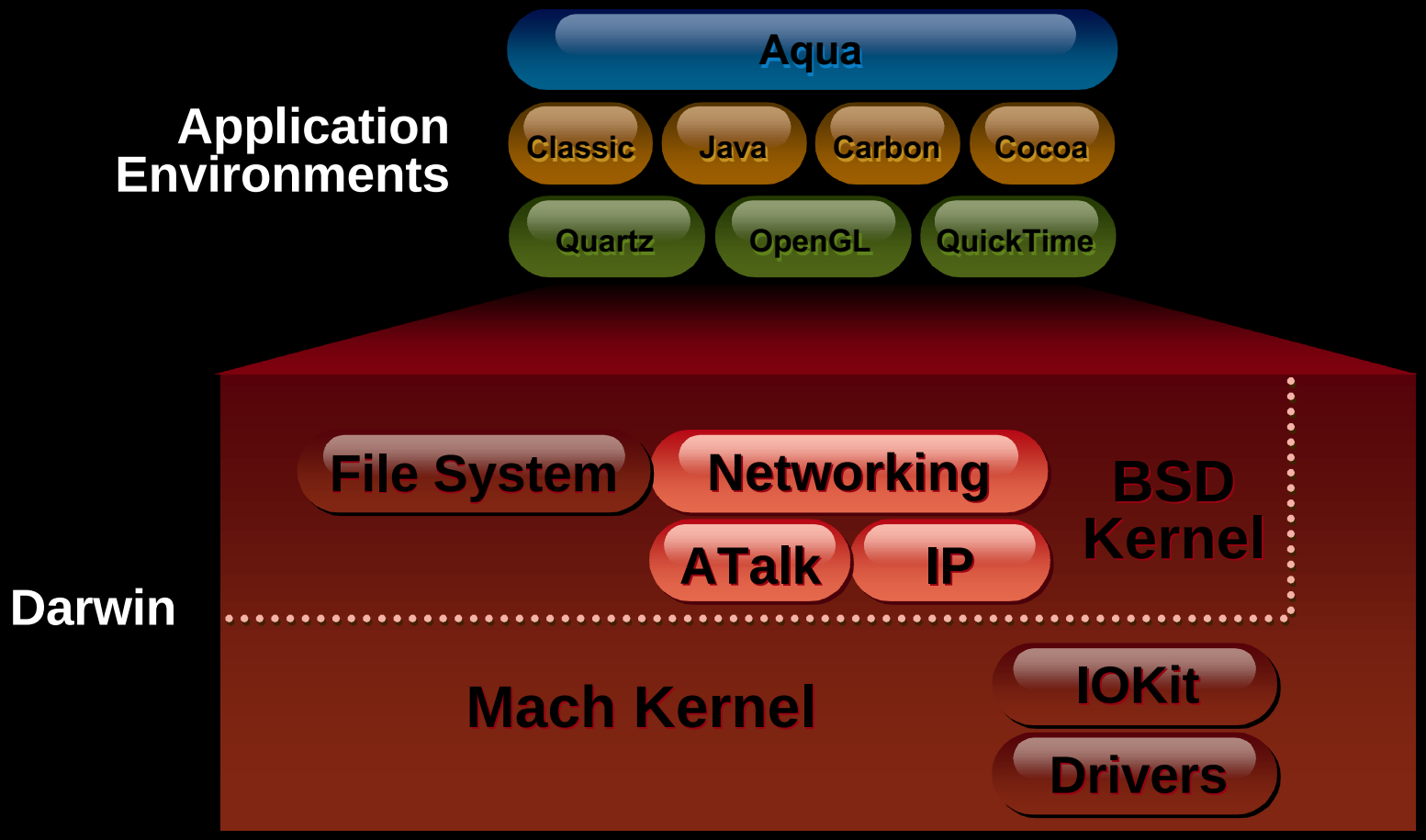
File System


Aqua
Application Environments
Classic
Java
Carbon
Cocoa
Quartz
OpenGL
QuickTime
File System
Networking
ATalk
IP
BSD Kernel
Darwin
Mach Kernel
IOKit
Drivers

File System


Aqua
Application Environments
Classic
Java
Carbon
Cocoa
Quartz
OpenGL
QuickTime
File System
Networking
BSD
Da
BSD with Extensions
Virtual File System
UFS
HFS
NFS
ISO
UDF
Etc.

Networking


Application Environments
Aqua
Classic
Java
Carbon
Cocoa
Quartz
OpenGL
QuickTime
Darwin
File System
Networking
ATalk
IP
BSD Kernel
Mach Kernel
IOKit
Drivers

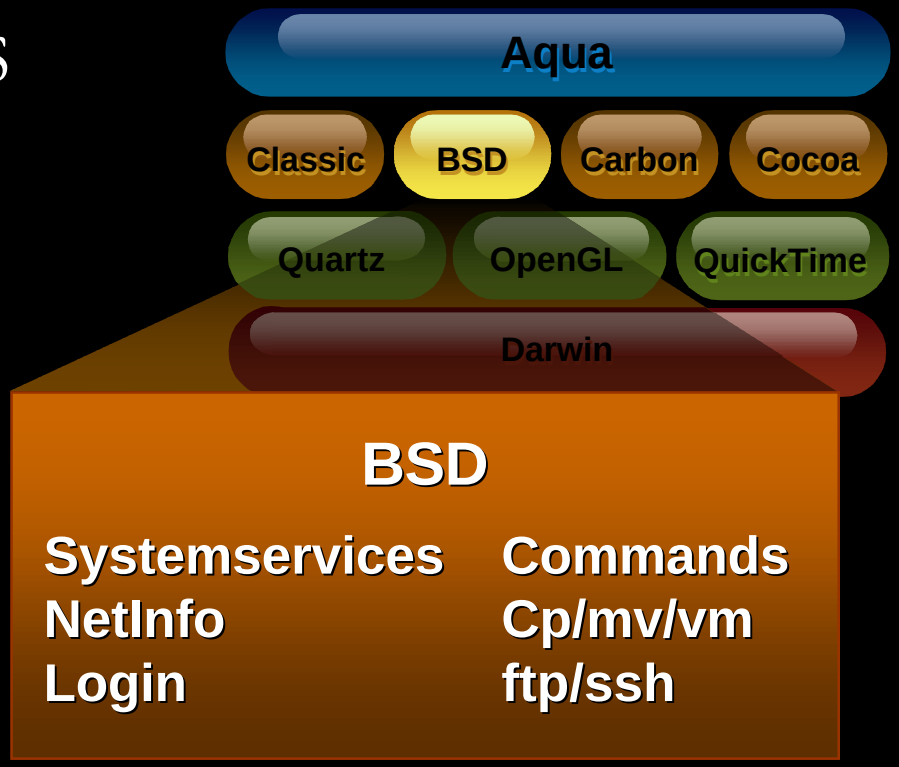
Roadmap

140 The Darwin Kernel

Civic
Fri., 3:30pm

BSD User Environment

- Command line/shells
- Network client tools
  - ssh, finger, ftp
- File tools
  - cp, ls, mkdir
- System tools
  - ps, kill, top, shutdown
- System services
  - NetInfo, DNS/BIND, NTP, syslog, Apache

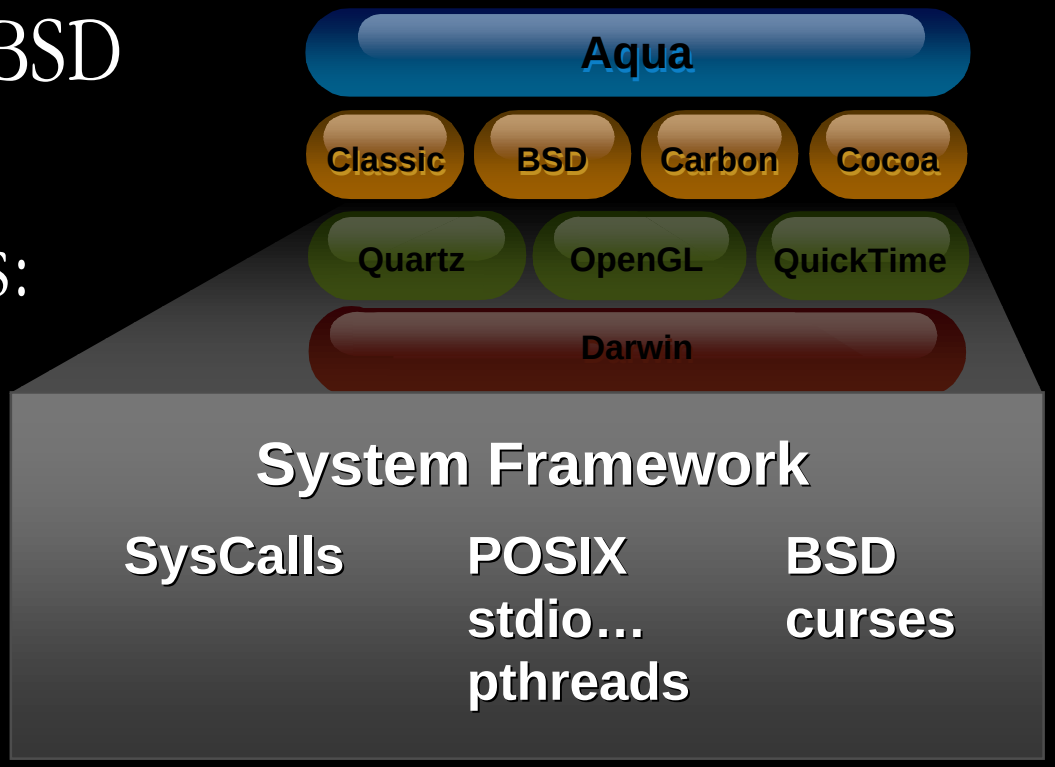


POSIX Applications

- Apache
- Emacs
- Sendmail
- Perl, Tcl, Python
- Samba

System Framework

- Exposes "POSIX"/BSD System APIs
- Additional services:
  - pthreads
  - Math library
  - C library



POSIX?

- Compliance is a goal
- Certification is not a goal
- The BSD API is a first-class citizen in Mac OS X
- Can co-exist with Carbon, Cocoa, and Java
- In fact, application environments build upon these APIs

When Should I Use the BSD API?

- Porting from POSIX
- Portability to POSIX
- When Carbon and Cocoa don't go down far enough
  - Use the highest level of abstraction

Porting From POSIX to Mac OS X

- No X Windows support
  - Can use 3rd-party X servers to help with porting
    - Tenon
    - XFree86
  - Cocoa is usually best choice
  - Carbon can work as well

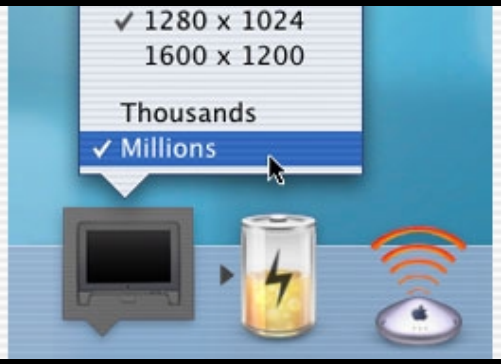
Porting From POSIX to Mac OS X

- Use -fno-common for dylibs
- <mach-o/dyld.h>, not dlopen()
- GNU make is the default
- autoconf usually works
  - powerpc-apple-darwin1.1

Where Things Go…

- /etc
- /bin, /sbin
- /var
- /tmp
- /usr
  - bin, sbin, libexec
  - share
  - lib
- /usr/local

Demo

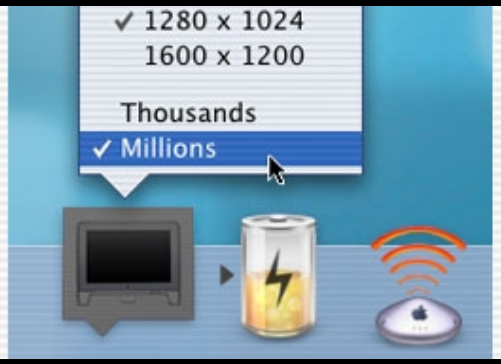


Eric Peyton
Engineer, Core OS Engineering

Why BSD?

- History
  - Tried-and-true
    - NeXTStep, OpenStep, Argon
    - Scalable
    - Robust
  - Well understood
- Open source
  - Active developer and user community

FreeBSD



Jordan K. Hubbard
Founder, Core Team Member, FreeBSD
Darwin Developer

FreeBSD Bundled Sources

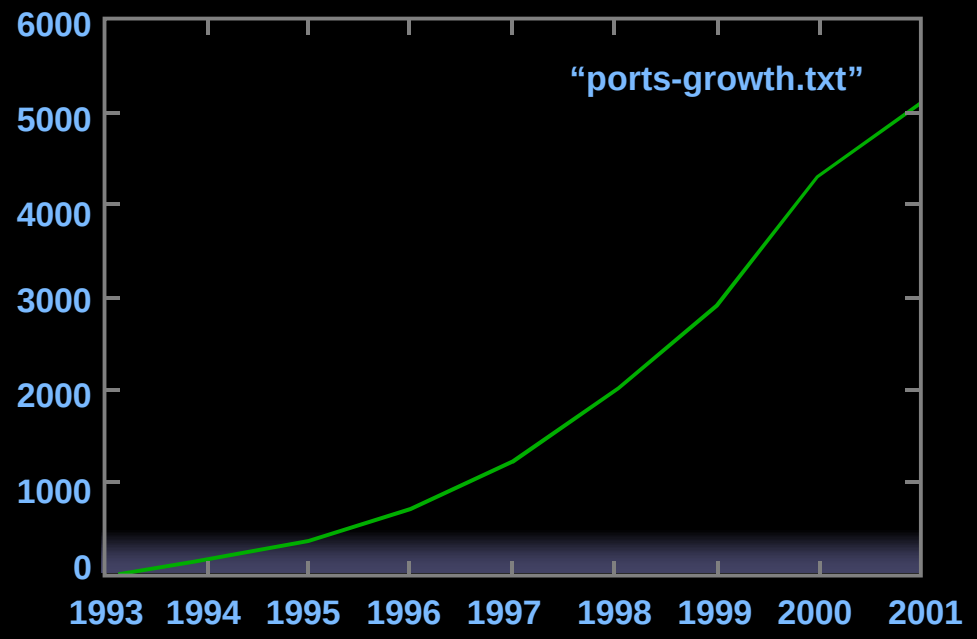
- /usr/src
  - Common Taxonomy
  - Vendor maintained software vs. project maintained software
  - Update, world and kernel targets
- Distributing the bits
  - CVSup: Understands branches and local repos
  - CTM: CVS bits through mail
  - AnonCVS: Standard remote interface
  - CVSWeb: See changes colorfully

FreeBSD Bundled Sources

- The multibranch development model, pros and cons
- Direct benefits to the Darwin community:
  - Userland creature comforts
  - Subsystems and security work
  - Driver reference material

FreeBSD 3rd-Party Apps

- The ports collection
  - Born August 20, 1994—by years end, contained 200 ports
  - 5103 ports in 52 categories today
  - Collection grows at over 50% a year



FreeBSD 3rd-Party Apps

- Standard Body—The OpenPackages project
- Ports collection features:
  - Automated fetch, configure, build, install and package dependency checking and simplistic conflict detection
  - Automated indexing and documentation
  - Binary packages
- What's really great about the ports collection
- What really sucks about the ports collection

The FreeBSD Development Community

- Run by 9 person Core team, elected from and by committers
- 267 CVS committers—docs, src, ports
- “Ownership”—collective vs. individual
- Hearts and minds vs. code—*BSD case studies
- The constantly evolving project identity—what drives it?

Where FreeBSD Can Work With Apple

- Shattering the desktop myth
- Capitalizing on Darwin's freedom to innovate

Wrapping It All Up…

- BSD is a powerful part of Mac OS X
- Avoid user exposure to UNIX
- Leverage Darwin
- Leverage FreeBSD
- Be part of the community

Getting More Information

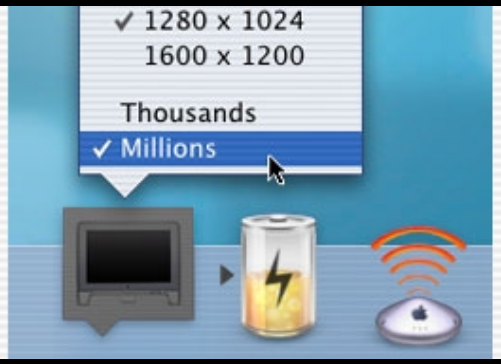
- Darwin
  http://www.opensource.apple.com/
- CAE Specification, Mac OS X/Open 1997
- Design and Implementation of the 4.4 BSD Operating System, McKusick, et. al.
- O'Reilly Books
  http://www.FreeBSD.org
  http://www.NetBSD.org

Roadmap

FF121 Feedback Forum: Darwin

Civic
Fri., 5:00pm

Q&A



Jason Yeo
Mac OS Technology Manager
jason@apple.com

http://developer.apple.com/events/urls.html

WWDC
Worldwide Developers Conference 2001